IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: ITIKARLAPALLI *et al*	
	Art Unit: 2168
Appl. No.: 10/709,522	
	Examiner: SANDERS,
Filed: 05/11/2004	AARON J
Simplifying Implementation of Custom Atomic Transactions in a Programming Environment	Attorney Docket No.: ORCL-003

Declaration by Inventor Under 37 CFR § 1.132

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

I, Krishnamohan Itikarlapalli, declare as follows:

1. I am one of the named inventors on the subject patent application.

2. My education includes a B.Tech Degree in Computer Science from Regional Engineering College, Warangal, India in June 1993.

3. Currently my work involves developing data access components and drivers for Oracle Database. The components are interfaces for different programming languages (like C, C++, Java, PHP etc.) to access Oracle database, the underlying resource management for high performance, scalability and security, data transfer protocol and transaction management. I had also developed earlier, Oracle applications for Energy sector using Oracle Designer, Forms, Reports and PL/SQL and an interpreter for the 4GL of a proprietary database- OASIS (Online Advanced Structured Information System).

4. Implementation of atomic transactions entails either completion of all the tasks (together performing the atomic transaction) or undoing (rolling back) the effects of any of the tasks already performed if all the tasks are not completed (i.e., abort the transaction).

5. The general need for atomic transactions has existed at least since 1980s, with the use of computers in areas such as banking.

6. One approach to providing the capability of implementation of atomic transactions entails a common framework implementing atomic transactions would be designed to automatically roll-back the effect of the completed tasks in case the corresponding atomic transaction is to be aborted.

7. Two of the references, Gostanian et al., U.S. 5,781,910 (hereafter "Gostanian") and Lordi et al., U.S. 5,857,204 (hereafter "Lordi"), relied upon by the Examiner fall within approach of point 6, noted above.

8. A problem with the common framework approach of point 6 is that it may not provide the customizations that different programs/users may wish in their specific applications, in case a transaction is to be aborted. These customizations are specified as a part of the user programs (which cannot be provided within the common framework shared by all user programs).

9. The need for such customizations of rollback procedures within user programs has been recognized for a long time and there have been several efforts in that regard in the industry.

10. The known closest prior art providing the ability to customize rollback procedures to programmers (i.e., within the user programs) is illustrated in Figure 1 (Applicants admitted prior art) of the subject patent application.

11. In the prior approach of Figure 1 of the subject patent application, the programmer is required to include program logic which keeps track of the specific tasks completed, and also to roll back the completed tasks (if the transaction is to be aborted).

12. The requirement of 11 makes design of complex atomic transactions (containing many tasks) extremely challenging for the programmers (implementing custom atomic transactions). For example, the programmer is required to keep track of the task procedures completed by the time of aborting, and to roll back such completed task procedures if the transaction is to be aborted. Line 152 of Figure 1 of the subject patent application shows a variable 'temp-was-one' being used to keep track of the completion of procedures P3(), P4() and P5(), which is then checked in line 170 before executing the corresponding roll-back procedures in the reverse order. It is particularly noted that the program code of lines 175-180 (including the roll-back procedures and the required order of execution) is expressly incorporated by the programmer (developer of user programs) and is a burden on the programmer developing the user program of Figure 1.

13. As the complexity of atomic transactions increases (thereby requiring execution of many tasks, depending on many different conditions), the prior approach noted above may not scale easily.

14. The need to simplify is integral to the software development arena, with implementation of custom atomic transactions in user programs not being an exception. The simplifications are generally limited by the available technologies and human creativity.

15. The disclosure of the present application provides programmers the ability to implement custom atomic transactions, while simplifying the challenges of points 11 - 14, noted above. In particular, the absence of keeping track of completed task procedures and the order of execution of the rollback procedures, is apparent by examining the user program of Figure 2, which is also implementing similar program logic as in Figure 1, but in an environment provided according to aspects of our invention. The program logic of lines 165-190 of Figure 1 has been simplified to a single Tabort call in line 280.

16. The advantage of point 15 in combination with the ability to implement custom rollback procedures (i.e., those specified in the user program) is not present in the known closest prior art, or in Gostanian and Lordi. The combination is important since customization provides the desired flexibility to the programmers, while the simplicity noted above makes it easier to

provide such customizations for complex atomic transactions. The independent claims (at least as sought to be amended) provide such a feature, as explained below.

17. Independent claim 1 (as currently amended) solves the challenges noted above in reciting a transaction manager which generates a transaction identifier, and a user program thereafter specifying combination of the transaction identifier, task procedure and a corresponding custom rollback procedure. On execution of a set of task procedures in a sequential order, the corresponding rollback procedures are kept track of, external to the user program, and the roll back procedures are executed in a reverse order of the sequential order if the atomic transaction is to be aborted. The programmer is relieved of the task of keeping track of the tasks completed, the corresponding order of execution, etc., while having the ability to specify custom rollback procedures.

18. Independent claim 7 (as currently amended) recites the operation of a user program, which specifies combinations of a transaction identifier (requested earlier from an external transaction manager), a task procedure and a corresponding roll-back procedure. Each combination indicates that the rollback procedure is to be executed if the execution of the corresponding task procedure in the combination is completed and if said atomic transaction is to be aborted. Thus, to abort a transaction after completion of execution of a set of task procedures, the user program merely needs to specify the transaction identifier associated with a abort procedure. The programmer thus has the ability to specify custom rollback procedures, in addition to having the benefit of simplicity due to the operation of the inventive features.

19. Independent claim 10 (as currently amended) recites the operation of a transaction manager that generates and provides a transaction identifier to a user program, and then receives combinations of the transaction identifier, a task procedure and a corresponding roll-back procedure from the user program. On execution of a set of task procedures in a sequential order, the corresponding rollback procedures are kept track of, external to the user program, and the roll back procedures are executed in a reverse order of the sequential order if the atomic transaction is to be aborted. Programmers of user programs thus have the ability to specify custom rollback procedures, in addition to having the benefit of simplicity due to the operation of the claimed transaction manager.

Reply to Final Office Action of 04/29/2009
Declaration by Inventor (37 CFR § 1.132): 08/29/2009

Appl. No.: 10/709,522
Atty. Docket No.: ORCL-003

20. Independent claim 16 (as currently amended) recites a system in which a user program specifies combinations of a transaction identifier (requested earlier from an external transaction manager), a task procedure and a corresponding roll-back procedure. On execution of a set of task procedures in a sequential order, the corresponding rollback procedures are kept track of, external to the user program, and the roll back procedures are executed in a reverse order of the sequential order if the atomic transaction is to be aborted.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Respectfully submitted,

Signature
Krishnamohan Itikarlapalli

Date: 31 st   August 2009